Amendments to the Specification:

Amend page 17, paragraph beginning at line 16, as follows:

In each of the above mentioned calendars, a pointer (a Flow ID) is used to represent a flow queue's location within the calendar. Thus, flow 0 has its Flow ID at 221 in calendar 220, flow 1 has its FlowID at 232 in calendar 230 and at 241 in the WFQ calendar 240 and flow 2047 has its FlowID at 231 in calendar 230 and at 251 in calendar 250, all as indicated by the arrows in Fig. 3. Further there may be none, one, or two such pointers to a single flow queue present in the plurality of calendars in the system. Typically, pointers in a calendar [[to]] do not represent un-initialized or empty flow queues. When a pointer to a flow queue (or a FlowID) is present in a particular calendar in the system, the flow queue may be referred to as being "in" that particular calendar.

Amend page 18, paragraph beginning at line 3, as follows:

Target port queues are control structures used to maintain ordered lists of frames that have common port destination and priorities. In the preferred embodiment, 2 priorities per media port (or output port) are provided to allow support of different classes of service, a so-called high priority target port queue and a so-called low priority target port queue. The selection of 2 priorities is a trade off between hardware cost and design complexity and is not intended to limit the scope of the invention. Further, the preferred embodiment includes a separate wrap queue 272 ~~270~~ and a discard port queue 270 ~~272~~.

Amend page 18, paragraph beginning at line 10, as follows:

Each of the time-based calendars 220, 230 and 250 consists of a plurality of epochs, with four shown for each in Fig. 3 as represented by the overlapping rectangles. Fig. 4 shows the four epochs 302, 304, 306 and 308 along with a typical timing arrangement for the epochs where the first epoch 302 (labeled epoch0) has a step of the scheduler tick (150 nsec in this case divided by 512 Bytes), the second epoch 304 has a step of 16 times that of the first epoch 302, with the third epoch 306 having the same ratio to the second epoch 304 and the fourth epoch 308 having the same ratio to the third epoch 306. In this way, the first epoch 302 has a high priority (it is scheduled for service sixteen times as often as the second epoch 304), creating a hierarchy of service priorities which will have associated increases in cost. A current pointer (e.g., 312 for epoch 302) is associated with each epoch to provide a pointer as to where in the ~~queue~~ calendar the processing is currently located. Since the arbitrary system of progressing through the epochs is to increment the current pointer, the direction of processing is from lower to higher in the epoch. Also shown in this Fig. 4 is the current time 320 and a scheduler tick 330 which drives the clock 320 as well as the priority selection.